

**PENGUJIAN OTOMATIS PADA WEBSITE
MANAJEMEN PRESENSI DAN PERHITUNGAN
TUNJANGAN APARATUR NEGARA
MENGUNAKAN CONTINUOUS INTEGRATION**

Tugas Akhir

**Diajukan untuk Memenuhi Salah Satu Persyaratan Mencapai Derajat
Sarjana Komputer**



Dibuat Oleh:

RHEZA HARYO HANGGARA

15 07 08576

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ATMA JAYA YOGYAKARTA**

2020

HALAMAN PENGESAHAN

Tugas Akhir Berjudul

PENGUJIAN OTOMATIS PADA WEBSITE MANAJEMEN PRESENSI DAN
PERHITUNGAN TUNJANGAN APARATUR NEGARA MENGGUNAKAN
CONTINUOUS INTEGRATION

yang disusun oleh

RHEZA HARYO HANGGARA

150708576

dinyatakan telah memenuhi syarat pada tanggal 08 Juli 2020

Dosen Pembimbing 1 : Stephanie Pamela Adithama, ST., MT.
Dosen Pembimbing 2 : Martinus Maslim, ST., MT.

Tim Penguji
Penguji 1 : Stephanie Pamela Adithama, ST., MT.
Penguji 2 : Yulius Harjoseputro, ST., MT.
Penguji 3 : Eduard Rusdianto, ST., MT.

Keterangan
Telah menyetujui
Telah menyetujui

Telah menyetujui
Telah menyetujui
Telah menyetujui

Yogyakarta, 08 Juli 2020

Universitas Atma Jaya Yogyakarta

Fakultas Teknologi Industri

Dekan

ttd

Dr. A. Teguh Siswanto, M.Sc

PERNYATAAN ORISINALITAS & PUBLIKASI ILMIAH

Saya yang bertanda tangan di bawah ini:

Nama Lengkap : Rheza Haryo Hanggara

NPM : 15 07 08576

Program Studi : Informatika

Fakultas : Teknologi Industri

Judul Penelitian : Pengujian Otomatis Pada Website Manajemen Presensi dan Perhitungan Tunjangan Aparatur Negara Menggunakan Continuous Integration

Menyatakan dengan ini:

1. Tugas Akhir ini adalah benar tidak merupakan salinan sebagian atau keseluruhan dari karya penelitian lain.
2. Memberikan kepada Universitas Atma Jaya Yogyakarta atas penelitian ini, berupa Hak untuk menyimpan, mengelola, mendistribusikan, dan menampilkan hasil penelitian selama tetap mencantumkan nama penulis.
3. Bersedia menanggung secara pribadi segala bentuk tuntutan hukum atas pelanggaran Hak Cipta dalam pembuatan Tugas Akhir ini.

Demikianlah pernyataan ini dibuat dan dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 8 Juli 2020

Yang menyatakan,

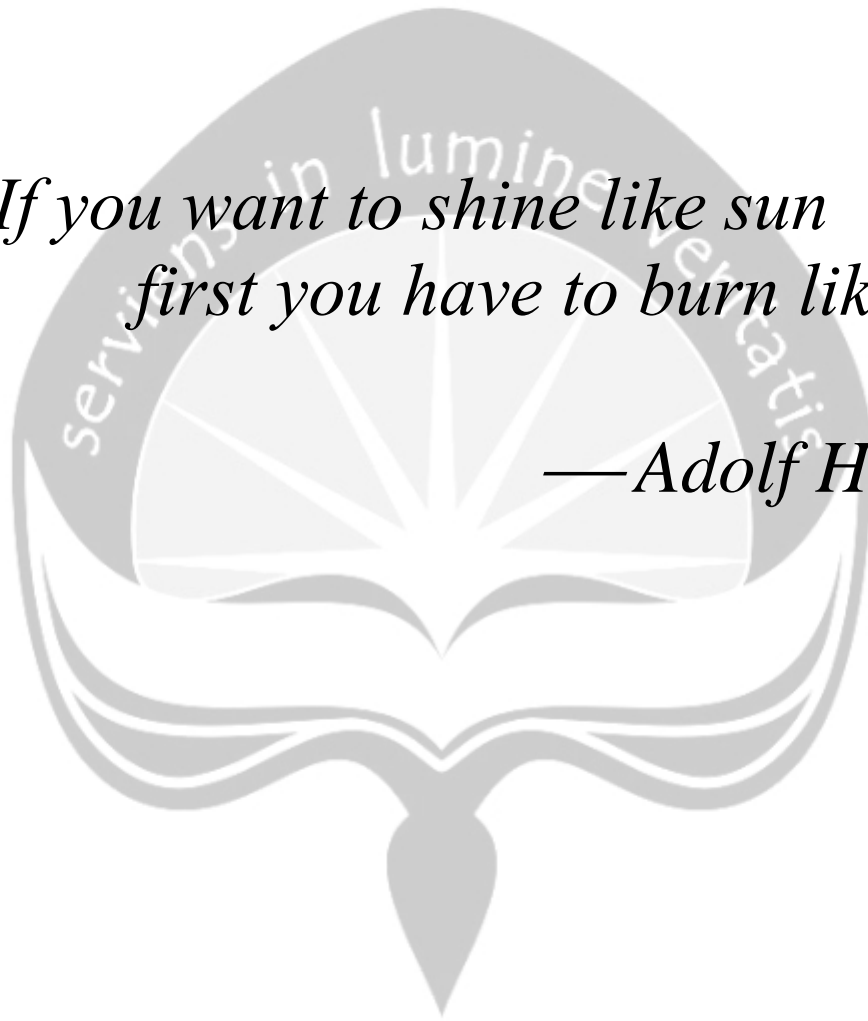
Rheza Haryo Hanggara

15 07 08576

HALAMAN PERSEMBAHAN

*“If you want to shine like sun
first you have to burn like it”*

—Adolf Hitler



KATA PENGANTAR

Puji syukur dan terima kasih senantiasa penulis panjatkan kepada Tuhan Yang Maha Esa karena Ia senantiasa memberikan nikmat, kasih sayang, dan karunia sehingga penulis dapat menyelesaikan penulisan tugas akhir ini. Rasa terima kasih penulis ucapkan juga kepada:

1. Allah SWT, Yang Maha Pengasih lagi Maha Penyayang, dengan limpahan nikmat, kasih sayang, dan karunia-Nya.
2. Bapak Dr. A. Teguh Siswanto M.Sc, selaku Dekan Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta.
3. Ibu Stephanie Pamela S.T., M.T., selaku dosen pembimbing I yang telah membimbing dan memberikan masukan serta motivasi kepada penulis untuk menyelesaikan tugas akhir ini.
4. Bapak Martinus Maslim S.T., M.T., selaku dosen pembimbing II yang telah membimbing dan memberikan masukan serta motivasi kepada penulis untuk menyelesaikan tugas akhir ini.
5. Mama, Papa, Adik, Pakde, Bude, Mas, Mbak serta keluarga besar lainnya yang mendukung penulis hingga saat ini.
6. Muhammad Rafi Arya Suta Pradana, Mochammad Deni Afrizal, Donny Dannu Wijaya, Denhaningrat Alensana, Billy Marcelino Alexander, Valentino Budiono, Hendy Adi Nugraha, Stanley Mae, Kurnia Dwi Nugroho dan Philip Abraham yang telah menjadi penyemangat, penghibur dan teman terdekat selama masa perkuliahan.
7. Dewa Gede Sukma Adhyaksa yang telah mendampingi dan membimbing penulis dari awal sampai tugas akhir ini diselesaikan.
8. Damar Mustiko Aji yang telah menjadi mentor serta membimbing penulis dalam mengerjakan tugas akhir ini.
9. Teman-teman Ralali.com yang mendukung serta membantu penulis berproses dalam pengembangan *soft skill* maupun *hard skill* sekaligus menjadi keluarga kecil baru selama penulis bekerja di Ralali.com

10. Teman-teman Mamikos.com yang mendukung serta membantu penulis berproses dalam pengembangan *soft skill* maupun *hard skill* sekaligus menjadi keluarga kecil baru selama penulis bekerja di Mamikos.com

Demikian laporan tugas akhir ini dibuat, dan penulis mengucapkan terima kasih banyak kepada semua pihak. Semoga laporan ini dapat bermanfaat bagi pembaca.

Yogyakarta, 8 Juli 2020

Rheza Haryo Hanggara

15 07 08576

DAFTAR ISI

PENGUJIAN OTOMATIS PADA WEBSITE MANAJEMEN PRESENSI DAN PERHITUNGAN TUNJANGAN APARATUR NEGARA MENGGUNAKAN CONTINUOUS INTEGRATION.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS & PUBLIKASI ILMIAH.....	ii
HALAMAN PERSEMBAHAN	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
INTISARI.....	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Metodologi	3
1.6 Sistematika Penulisan Tugas Akhir.....	5
BAB II TINJAUAN PUSTAKA.....	6
BAB III LANDASAN TEORI.....	9
3.1 Pengujian Perangkat Lunak	9
3.2 Test Case.....	9
3.3 Integration Testing	10
3.4 Automated Testing.....	10
3.5 Continuous Integration	11
3.6 Mocha.js.....	13
3.7 Chai.js	14
3.8 Javascript	14
BAB IV ANALISIS DAN PERANCANGAN	15
4.1 Deskripsi Masalah.....	16
4.2 Analisis Kebutuhan Pengujian.....	16
4.2.1 Analisis Pengujian Manual.....	16
4.2.2 Analisis Pengujian Otomatis.....	16
4.3 Perancangan Test Case.....	17

4.3.1 Otentikasi Pengguna.....	18
4.3.2 Menampilkan Data Organisasi Perangkat Daerah (OPD).....	20
4.3.3 Mencari Data Organisasi Perangkat Daerah (OPD).....	21
4.3.4 Menambah Data Organisasi Perangkat Daerah (OPD).....	23
4.3.5 Menyunting Data Organisasi Perangkat Daerah (OPD).....	26
4.3.6 Menghapus Data Organisasi Perangkat Daerah (OPD).....	29
4.3.7 Menampilkan Data Eselon.....	30
4.3.8 Mencari Data Eselon.....	31
4.3.9 Menambah Data Eselon.....	33
4.3.10 Menyunting Data Eselon.....	36
4.3.11 Menyunting Data Eselon.....	39
4.3.12 Menampilkan Data Golongan.....	39
4.3.13 Mencari Data Golongan.....	40
4.3.14 Menambah Data Golongan.....	42
4.3.15 Menyunting Data Golongan.....	45
4.3.16 Menghapus Data Golongan.....	47
4.3.17 Menampilkan Data Aparatur Sipil Negara (ASN)	48
4.3.18 Mencari Data Aparatur Sipil Negara (ASN)	49
4.3.19 Menambah Data Aparatur Sipil Negara (ASN)	51
4.3.20 Menyunting Data Aparatur Sipil Negara (ASN)	55
4.3.21 Menghapus Data Aparatur Sipil Negara (ASN)	59
4.3.22 Menampilkan Data Kalender Kerja.....	60
4.3.23 Menampilkan Data Kalender Kerja.....	61
4.3.24 Menambah Data Kalender Kerja.....	63
4.3.25 Menyunting Data Kalender Kerja.....	67
4.3.26 Menghapus Data Kalender Kerja.....	70
4.3.27 Menampilkan Data Shift Kerja.....	71
4.3.28 Mencari Data Shift Kerja.....	72
4.3.29 Menambah Data Shift Kerja.....	74
4.3.30 Menyunting Data Shift Kerja.....	76
4.3.31 Menghapus Data Shift Kerja.....	79
4.3.32 Menampilkan Data Tambahan Penghasilan Pegawai (TPP)	80

4.3.33 Mencari Data Tambahan Penghasilan Pegawai (TPP).....	81
4.3.34 Menambah Data Tambahan Penghasilan Pegawai (TPP)	83
4.3.35 Menyunting Data Tambahan Penghasilan Pegawai (TPP)	86
4.3.36 Menghapus Data Tambahan Penghasilan Pegawai (TPP)	88
4.3.37 Menampilkan Data Pengguna.....	89
4.3.38 Mencari Data Pengguna.....	90
4.3.39 Menambah Data Pengguna.....	92
4.3.40 Menyunting Data Pengguna.....	95
4.3.41 Menghapus Data Pengguna.....	99
BAB V. IMPLEMENTASI DAN PENGUJIAN SISTEM.....	98
5.1 Deskripsi Pengujian.....	100
5.1.1 Alur Pengujian Secara Otomatis.....	100
5.2 Analisis Hasil Pengujian Otomatis.....	101
5.2.1 Penerapan <i>Automated Testing</i> Pada <i>Mocha</i> dan <i>Cypress</i> Untuk Test Case Otentikasi.....	101
5.2.2 Penerapan <i>Automated Testing</i> Pada <i>Mocha</i> dan <i>Cypress</i> Untuk Test Case OPD (Organisasi Perangkat Daerah).....	103
5.2.3 Penerapan <i>Automated Testing</i> Pada <i>Mocha</i> dan <i>Cypress</i> Untuk Test Case Golongan.....	117
5.2.4 Penerapan <i>Automated Testing</i> Pada <i>Mocha</i> dan <i>Cypress</i> Untuk Test Case ASN (Aparatur Sipil Negara).....	127
5.2.5 Penerapan <i>Automated Testing</i> Pada <i>Mocha</i> dan <i>Cypress</i> Untuk Test Case Eselon.....	144
5.2.6 Penerapan <i>Automated Testing</i> Pada <i>Mocha</i> dan <i>Cypress</i> Untuk Test Case Kalender Kerja.....	157
5.2.7 Penerapan <i>Automated Testing</i> Pada <i>Mocha</i> dan <i>Cypress</i> Untuk Test Case Shift Kerja.....	172
5.2.8 Penerapan <i>Automated Testing</i> Pada <i>Mocha</i> dan <i>Cypress</i> Untuk Test Case TPP (Tambahan Penghasilan Pegawai)	187
5.2.9 Penerapan <i>Automated Testing</i> Pada <i>Mocha</i> dan <i>Cypress</i> Untuk Test Case Pengguna.....	199
5.2.10 Penerapan <i>Continuous Integration</i>	213
5.1 Pembahasan Hasil Pengujian <i>Automated Testing</i> Dengan <i>Mocha</i> dan <i>Cypress</i> Menggunakan <i>Continuous Integration</i>	216
BAB VI. PENUTUP.....	218
6.1 Kesimpulan.....	221
6.2 Saran.....	221
DAFTAR PUSTAKA.....	220

INTISARI

PENGUJIAN OTOMATIS PADA WEBSITE MANAJEMEN PRESENSI DAN PERHITUNGAN TUNJANGAN APARATUR NEGARA MENGGUNAKAN CONTINUOUS INTEGRATION

Rheza Haryo Hanggara
15 07 08576

Dewasa kini kian meningkat penggunaan *Continuous Integration* (CI) yang dilakukan untuk mempermudah serta mempercepat proses pengunggahan kode. CI merupakan praktik mengotomatiskan integrasi perubahan kode dari banyak kontributor ke proyek perangkat lunak. Salah satu keuntungan teknis dari CI yaitu mengintegrasikan potongan-potongan kecil kode pada satu waktu. CI juga meningkatkan reliabilitas pengujian dengan adanya perubahan spesifik terhadap system serta memungkinkan untuk melakukan pengujian positif dan negatif secara akurat.

Pada tugas akhir ini akan dilakukan pengujian terhadap *Synchrone* secara otomatis. Dalam pengujian tersebut terdapat 2 *layer* pengujian yang berbeda, yaitu: *layer* API (*Application Programming Interface*) atau *layer service* dan *layer* UI (*User Interface*) atau *layer* Antarmuka. Pengujian akan dilakukan pada seluruh fungsi utama dari *Synchrone*. Pengujian ini nantinya akan membuktikan efisiensi waktu yang dilakukan oleh pengujian otomatis, agar kedepannya pengujian otomatis dapat mengurangi beban waktu pengujian manual.

Berdasarkan pengujian yang dilakukan terhadap *Synchrone*, ditemukan *bug* di 8 fungsi utama yang diuji. Setelah melakukan perhitungan terhadap pengujian otomatis API dan UI, didapatkan rata-rata waktu tempuh sebesar 52,5 detik dan 2 menit 6 detik. Dengan rata-rata waktu tempuh tersebut, dapat dikatakan bahwa pengujian otomatis API dan UI sangatlah efisien. Dengan berhasilnya mengimplementasikan *Continuous Integration*, juga mempermudah dan mempercepat proses pengunggahan kode dari pengembang perangkat lunak. Dengan hasil tersebut, dapat disimpulkan bahwa pengujian otomatis lebih efisien jika diukur berdasarkan waktu tempuh pengujian.

Kata kunci: Pengujian Otomatis, *Synchrone*, Continuous Integration, *API Testing*, *UI Testing*.

Dosen Pembimbing I : Stephanie Pamela Adithama S.T., M.T.

Dosen Pembimbing II : Martinus Maslim, S.T., M.T.

Jadwal Sidang Tugas Akhir : 8 Juli 2020

BAB I PENDAHULUAN

1.1 Latar Belakang

Dewasa ini kian meningkat penggunaan *Continuous Integration* (CI) untuk mempermudah serta mempercepat proses pengunggahan kode. Salah satu keuntungan teknis dari CI yaitu mengintegrasikan potongan-potongan kecil kode pada satu waktu. CI juga meningkatkan reliabilitas pengujian dengan adanya perubahan spesifik terhadap system serta memungkinkan untuk melakukan pengujian positif dan negatif secara akurat. Penggunaan CI sangat berdampak pada pembangunan perangkat lunak, dimana hal tersebut tentunya dapat meningkatkan reliabilitas sistem.

Website Synchome berguna untuk mengurangi tingkat ketidakdisiplinan PNS dan juga mengatur tentang Tambahan Penghasilan Pegawai (TPP). PNS akan menerima TPP yang pembayarannya diatur menurut jabatan dan golongan, serta tingkat kehadiran. Tujuan dari TPP ini untuk menyedjahterakan dan memotivasi PNS untuk mendukung pelaksanaan tugas. Besaran TPP ini akan berbeda untuk setiap jabatan dan golongan. Untuk jabatan tertinggi Jabatan Pimpinan Tinggi Madya, besaran TPP-nya sebesar Rp 30.000.000 per bulannya, sedangkan untuk jabatan terendah Pelaksana Golongan I sebesar Rp 2.400.000. Besaran TPP ini akan dihitung lagi secara proporsional berdasarkan jumlah kehadiran dan ketidakhadiran dengan kondisi tertentu dan dibagi dengan jumlah hari kerja dalam satu bulan. PNS yang kerap tidak hadir tanpa keterangan yang jelas, akan mendapatkan denda potongan dari TPP-nya [1]. *Website Synchome* sudah dibangun namun belum diluncurkan. Terdapat beberapa tahap yang belum dilakukan, salah satunya adalah pengujian.

Dalam pembangunan *website*, seringkali dilakukan pengujian perangkat lunak atau *software testing* sebelum dilakukannya peluncuran *website* tersebut. Pengujian perangkat lunak adalah suatu proses atau serangkaian proses yang dirancang untuk meyakinkan kode komputer telah melakukan tugasnya dan

tidak melakukan selain tugas tersebut [2]. Dengan dilakukannya *testing*, dapat mengurangi intensitas *bug* dalam suatu software. Terdapat banyak metode untuk melakukan software testing yaitu: *unit testing*, *performance testing*, *integration testing*, *user acceptance test* dan lain-lain. Dengan adanya perkembangan teknologi semakin banyak cara untuk melakukan *testing*. *Software testing* telah menjadi lebih sulit dan lebih mudah dari sebelumnya [2].

Dari semua metode yang ada *testing* cenderung dilakukan secara manual. *Testing* dengan cara manual akan memakan waktu yang cukup lama. Terdapat sebuah solusi untuk mempercepat proses *testing* dengan menggunakan *automation test*. *Automation test* adalah sebuah test yang dilakukan dengan cara menuliskan perintah-perintah yang kemudian dijalankan secara otomatis. Setelah diterapkan, *automation test* umumnya jauh lebih ekonomis, biaya menjalankannya hanya sebagian kecil dari upaya untuk melakukan manual *testing* [3]. *Automation testing* melakukan banyak pengujian dalam skala tertentu, namun biasanya jika melakukan *automation testing* maka di dalam skala tersebut terdapat *integration testing*. *Integration testing* adalah pengujian yang dilakukan untuk mengekspos kecacatan yang ada pada antarmuka dan di dalam interaksi antara komponen atau sistem yang terintegrasi.

Metode ini terus digunakan dalam melakukan testing manual ataupun menggunakan *automation testing*. Kemudian *integration testing* berkembang menjadi *continuous integration testing*. CI adalah praktik pengembangan perangkat lunak dimana anggota tim mengintegrasikan pekerjaan mereka, biasanya setiap orang berintegrasi setidaknya setiap hari, berdasarkan beberapa integrasi per hari [4].

Penelitian ini ditujukan untuk mempermudah pengembang perangkat lunak pada saat melakukan pengujian. Pengembang perangkat lunak biasanya menggunggah kode mereka secara bersamaan dan kode tersebut akan diuji satu per satu. Dengan penelitian ini, pengujian akan dilakukan secara bersamaan sesuai waktu yang telah ditentukan. Sehingga pengembang perangkat lunak tidak harus menunggu untuk proses pengujian.

1.2 Rumusan Masalah

Berdasarkan penjelasan yang terdapat pada latar belakang, permasalahan yang dapat diangkat dalam penelitian ini adalah bagaimana cara agar mempercepat proses pengujian website dengan membangun perangkat pengujian yang efisien serta menggunakan *Continuous Integration*.

1.3 Tujuan

Tujuan dari penelitian yang ingin dicapai yaitu dapat membangun perangkat pengujian yang efisien serta dapat mengimplementasikan *Continuous Integration*.

1.4 Batasan Masalah

Terdapat keterbatasan bagi penulis dalam menyelesaikan penelitian ini maka dari itu penulis membuat beberapa Batasan, yaitu:

1. Penelitian ini hanya menggunakan *Integration Testing* sebagai metode pengujian.
2. Penelitian ini ditujukan untuk mengatasi permasalahan yang ada dalam website *Synchrome*.

1.5 Metodologi

Adapun metodologi yang digunakan oleh penulis untuk menjabarkan proses dari pembuatan Tugas Akhir untuk mendapatkan hasil yang diinginkan, yaitu:

1. Studi Literatur

Tahap pertama dilakukan untuk mencari referensi yang akan digunakan oleh penulis, seperti: buku, jurnal dan sumber lain untuk mempertegas teori-teori yang dilakukan.

2. Wawancara

Tahap kedua dilakukan dengan pengembang *website Synchrome* untuk memperoleh data-data yang mendukung dalam proses pembuatan sistem.

3. Analisis Kebutuhan

Tahap ketiga dilakukan untuk mengetahui sistem, baik kebutuhan fungsional ataupun nonfungsional.

4. Perancangan Perangkat Pengujian

Tahap keempat dilakukan agar perangkat pengujian yang nantinya akan dibangun sesuai dengan cakupan, serta penggunaan metode pengujian yang telah disesuaikan berdasarkan kebutuhan sistem.

5. Pembangunan Perangkat Pengujian

Tahap kelima dilakukan setelah semua data-data pendukung terkumpul dengan menggunakan *Software Testing Life Cycle* (STLC). Pembangunan sistem meliputi pengkodean untuk merealisasikan sistem sesuai dengan yang diharapkan oleh penulis.

6. Implementasi Perangkat Pengujian

Tahap keenam dilakukan untuk mengetahui kekurangan dan kelebihan dari sistem, baik fungsional atau nonfungsional.

7. Penulisan Laporan

Tahap ketujuh dan terakhir dilakukan untuk mendokumentasikan segala proses yang terjadi selama proses pembuatan Tugas Akhir agar nantinya dapat dimanfaatkan dan dikembangkan oleh peneliti dengan topik serupa.

1.6 Sistematika Penulisan Tugas Akhir

Secara sistematis, penulisan laporan akhir disusun sebagai berikut:

BAB I – Pendahuluan

Bab ini berisi tentang uraian tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, metodologi dan sistematika penulisan laporan dari Tugas Akhir

BAB II – Tinjauan Pustaka

Bab ini berisi tentang uraian mengenai penelitian terdahulu dengan topik yang serupa yang akan ditinjau dan digunakan oleh penulis sebagai pembanding dan acuan dalam pembahasan masalah.

BAB III – Landasan Teori

Bab ini berisi tentang dasar-dasar teori yang digunakan oleh penulis sebagai data pendukung, pedoman dan acuan dalam merumuskan masalah hingga membangun sistem

BAB IV – Analisis dan Perancangan Pengujian

Bab ini berisi tentang analisis serta proses perancangan sistem, seperti: *Test Case*, *User Acceptance Criteria* dan, *Application Programming Interface* (API).

BAB V – Implementasi dan Pengujian Sistem

Bab ini berisi tentang pembahasan penerapan sistem yang dibuat mulai dari implementasi hingga pengujian perangkat lunak.

BAB VI – Kesimpulan dan Saran

Bab ini merupakan penutup yang terdiri atas kesimpulan dan saran yang didapat selama pembuatan Tugas Akhir guna untuk pembelajaran atau pengembangan lebih lanjut di kemudian hari.

BAB II TINJAUAN PUSTAKA

Pada penelitian sebelumnya, Milad Hanna, Amal Elsayed Aboutbl, Mostafa-Sami M. Mostafa telah melakukan penelitian tentang “Automated Software Testing Framework for Web Application”. Penelitian ini membandingkan efesiensi waktu dalam pengujian menggunakan *Traditional Automation*, *Selenium IDE*, *SAT Framework*. Hasil dari penelitian ini, pengujian tercepat berhasil dicapai menggunakan *SAT Framework*, dengan waktu tempuh 182 menit. Tujuan utama dari penelitian ini adalah untuk mengusulkan perangkat pengujian yang baru. Dimana perangkat tersebut diharapkan dapat memberikan lebih banyak bantuan untuk penguji dalam keseluruhan proses pengujian otomatis perangkat lunak. Kerangka yang diusulkan dapat berhasil digunakan untuk mengotomatiskan proses pembuatan skrip pengujian. Kerangka Kerja SAT (*Standart Assesment Task*) yang diusulkan bermanfaat khususnya ketika SUT (*System Under Test*) [5].

Penelitian lain juga dilakukan oleh M. Sidi Mustaqbal, Roeri Fajri Firdaus, Hendra Rahmadi telah melakukan penelitian tentang “Pengujian Aplikasi Menggunakan Blackbox Testing Boundary Value Analysis”. Penelitian ini menggunakan boundary value analysis dengan menentukan batas bawah dan batas atas dari data yang ada. Hasil dari penelitian ini, pengujian menggunakan metode *Boundary Value Analysis* perlu dilengkapi dengan beberapa proses validasi data untuk menjamin akurasi entri data agar sesuai dengan fungsionalitas yang diinginkan. Peneliti menyimpulkan bahwa metode *Black Box Testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang diharapkan. Kemudian estimasi data uji dapat dihitung melalui formulir masukkan data yang akan diuji, aturan masukkan data harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dengan menggunakan *Integration Test* sebagai penengah [6].

Kemudian, penelitian selanjutnya dilakukan oleh Danang Wahyu Utomo, Defri Kurniawan, Yani Parti Astuti telah melakukan penelitian tentang “Teknik Pengujian Perangkat Lunak Dalam Evaluasi Sistem Layanan Mandiri Pemantauan Haji Pada Kementrian Agama Provinsi Jawa Tengah”. Penelitian ini menggunakan metode pengujian blackbox, whitebox, dan *user acceptance testing* dalam upaya meminimalisir bug yang terdapat pada sistem. Hasil dari penelitian ini, peneliti berhasil meminimalisir error ataupun bug yang terdapat pada sistem. Kemudian peneliti menyarankan agar menggunakan metode *Black box*, *White box*, dan *User Acceptance Testing*. Pengujian *White Box* berperan sebagai teknik pengujian terhadap kode program yang melibatkan para pembangun sistem untuk mengetahui kompleksitas dari program. Pengujian *Black Box* disini berperan sebagai tempat pengujian eksternal sistem yang melibatkan *stakeholder*, agar nantinya dapat mengetahui masukan dan luaran sudah sesuai dengan hasil yang diharapkan dan juga dapat menampung kritik dan saran yang diberikan oleh *stakeholder* [7].

Berdasarkan dari ketiga penelitian tersebut dapat dilihat, bahwa dalam melakukan pengujian dapat dilakukan dengan menggunakan beberapa metode serta *tools* yang berbeda. Dengan metode dan *tools* yang berbeda, para peneliti terdahulu mendapatkan hasil yang berbeda-beda. Dari hasil tersebut dapat dikatakan bahwa melakukan pengujian otomatis memberikan pengaruh yang sangat besar dalam perbedaan waktu pengujian maupun sumber daya manusia yang digunakan. Melalui ketiga penelitian tersebut, penulis akan mencoba melakukan pengujian dengan menggunakan *Cypress* sebagai *framework testing* dan melakukan *Automated Testing* dan *Continuous Integration Testing*.

Adapun kelebihan dari penelitian ini yaitu perangkat pengujian dapat meminimalisir *bug*, mempercepat proses pengujian dan meningkatkan performa *website*. Dengan adanya *Continuous Integration* juga dapat membantu pengembang perangkat lunak agar tidak menunggu untuk proses pengujian yang dilakukan.

Tabel 2.1 – Perbandingan Penelitian

Peneliti	M. Hanna, A. E. Aboutbl, M. Mostafa [5]	M. S. Mustaqbal, R. F. Firdaus, H. Rahmadi [6]	D. W. Utomo, D. Kurniawan, Y. P. Astuti [7]	Hanggara, R. Haryo (2020)*
Judul Penelitian	Automated Software Testing Framework for Web Application	Pengujian Aplikasi Menggunakan Blackbox Testing Boundary Value Analysis	Teknik Pengujian Perangkat Lunak Dalam Evaluasi Sistem Layanan Mandiri Pemantauan Haji Pada Kementerian Agama Provinsi Jawa Tengah	Automated Testing dan Continuous Integration Pada Website Manajemen Presensi dan Perhitungan Tunjangan Aparatur Sipil Negara di Provinsi Bali
Metode	<i>Black Box</i>	<i>Boundary Value Analysis</i>	<i>Black Box, White Box, dan User Acceptance Test</i>	<i>Gray Box</i>
Alat	<i>Traditional Automation, Selenium IDE dan SAT Framework</i>	-	<i>Hypertext Preprocessor (PHP)</i>	<i>Cypress, Mocha JS dan Chai JS</i>
Objek Pengujian	<i>Web App</i>	Aplikasi Prediksi Kelulusan SNMPTN	Sistem Layanan Mandiri Haji Kemenag Provinsi Jawa Tengah	Website Manajemen Presensi dan Perhitungan Tunjangan Aparatur Sipil Negara di Provinsi Bali

*Penelitian yang dilakukan

BAB VI. PENUTUP

6.1 Kesimpulan

Dari hasil penelitian yang telah dipaparkan pada sub bab sebelumnya dapat diambil kesimpulan, antara lain:

Telah berhasil membangun perangkat pengujian yang efisien serta dapat mengimplementasikan *Continuous Integration*. Pada pengujian otomatis menggunakan *Mocha* dan *Cypress* terbilang efisien jika diukur dari waktu tempuh pengujian yang menguji 41 fungsionalitas utama dari *Synchrome*. Dalam melakukan pengujian otomatis API (*Application Programming Interface*) setiap fungsionalitasnya membutuhkan waktu kurang dari 1 detik. Waktu tempuh pengujian otomatis dapat dipengaruhi beberapa faktor, yaitu: koneksi internet, RAM (*Random Access Memory*) dari perangkat pengujian dan objek dari *Synchrome*. Dengan diimplementasikannya *Continuous Integration* pengembang tidak perlu lagi untuk menunggu proses pengujian yang dilakukan oleh penguji di mesin pribadi mereka, karena proses pengujian sudah dilakukan oleh *Continuous Integration* itu sendiri.

6.2 Saran

Berdasarkan dari penelitian yang telah dilakukan, adapun saran-saran yang dapat dijadikan sebagai bahan evaluasi untuk pengembangan dan penelitian selanjutnya, sebagai berikut:

1. Untuk penelitian selanjutnya disarankan untuk membuat standarisasi objek pengujian yang digunakan untuk pengujian UI (*User Interface*).
2. Standarisasi untuk dokumentasi API (*Application Programming Interface*) atau TRD (*Technical Requirements Document*) agar proses perancangan API dan pengujian otomatis API dapat berjalan secara paralel.

3. Untuk mengurangi waktu tempuh dalam pengujian otomatis, sebaiknya menyediakan *server* dan konektivitas yang stabil.



DAFTAR PUSTAKA

- [1] J. Adhyaksa, *Pembangunan Aplikasi Manajemen Presensi dan Perhitungan Tunjangan Aparatur Negara. Tugas Akhir. Universitas Atma Jaya Yogyakarta. Yogyakarta. 2019.*
- [2] G. J. Myers, C. Sandler, and T. Badgett, *The Art of Software Testing 3rd Edition*. New Jersey: John Wiley & Sons Inc, 2011.
- [3] F. Mark, G. Dorothy, *Software Test Automation: Effective Use of Test*. New York: Addison Wesley Professional, 1999.
- [4] P. M. Duvall, S. Matyas, and A. Glover, *Continuous Integration: Improving Software Quality and Reducing Risk*. New Jersey: Addison Wesley Professional, 2007.
- [5] M. Hanna, A. E. Aboutabl, and M. M. Mostafa, "Automated Software Testing Framework for Web Applications", *International Journal of Applied Engineering Research*. vol. 13, no. 11, pp. 9758–9767, 2018.
- [6] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, "Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN)", *Jurnal Ilmiah Teknologi Informasi Terapan*. vol. I, no. 3, pp. 31–36, 2015.
- [7] D. W. Utomo et al., "Teknik pengujian perangkat lunak dalam evaluasi sistem layanan mandiri pemantauan haji pada kementerian agama provinsi jawa tengah", *Jurnal SIMETRIS*. vol. 9, no. 2, pp. 731–746, 2018.
- [8] B. R. Patton, *Software Testing Publisher : Sams Publishing Pub Date : July 26 , 2005 ISBN : 0-672-32798-8 Pages : 408 Software Testing By Ron Patton Publisher : Sams Publishing Pub Date : July 26 , 2005 ISBN : 0-672-32798-8 Pages : 408*. 2005.
- [9] E. Vander Veer, *JavaScript for Dummies 4Th Ed*. Indianapolis: Wiley

Publishing Inc. 2005.

[10] MochaJS Team, (2011, November 22). Introduction to Mocha JS [Online].

Available: <https://mochajs.org>.

[11] ChaiJS Team, (2011, Desember 07). *Introduction to ChaiJS* [Online].

Available: <https://www.chaijs.com/>

